**Basic Variable Types**

**Solidity Types:**

Solidity supports various elementary types that can be combined to create more complex ones. You can read more about them in the Solidity documentation (<https://docs.soliditylang.org/en/v0.8.20/types.html#types>).

For now, let’s focus on the most commonly used

* Boolean (bool): true or false
* Unsigned Integer (uint): unsigned whole number (positive)
* Integer (int): signed whole number (positive and negative)
* Address (address): 20 bytes value. An example of an address can be found within your MetaMask account.
* Bytes (bytes): low-level raw byte data

**Variables definition:**

Variables are just placeholders for values. A value can be one data type described in the list above. For instance, we could create a Boolean variable named `hasFavoriteNumber`, which would represent whether someone has a favourite number or not (constant `true` or `false`).

Bool hasFavoriteNumber = true; // The variable `hasFavoriteNumber` represents the value `true`

It’s possible to specify the number of bits used for `uint` and `int`. For example, uint256 specifies that the variable has 256 bits. Uint is a shorthand for uint256.

**Note:** It’s always advisable to be explicit when specifying the length of the data type.

The semicolon at the end of each line signifies that a statement is completed.

// SPDX-License-Identifier: MIT

Pragma solidity 0.8.19;

Contract SimpleStorage {

// Basic types

bool hasFavoriteNumber = true;

uint256 favoriteNumber = 88;

string favoriteNumberInText = “eighty-eight”;

int256 favoriteInt = -88;

address myAddress = 0xaB1B7206AA6840C795aB7A6AE8b15417b7E63a8d;

bytes32 favoriteBytes32 = “cat”;

}

**Bytes and strings:**

Bytes are a collection of characters written in hexadecimal representation.

bytes1 minBytes = “I am a fixed size byte array of 1 byte”;

bytes32 maxBytes = “I am a fixed size byte array of 32 bytes”;

bytes dynamicBytes = “I am a dynamic array, so you can manipulate my size”;

Bytes can be allocated in size (up to `bytes32`). However, bytes and bytes32 represent distinct data types.

**Strings** are internally represented as dynamic byte arrays (`bytes` type) and designed specifically for working with text. For this reason, a string can easily be converted into bytes.

Bits and Bytes overview (<https://www.youtube.com/watch?v=Dnd28lQHquU>)

**The contract logic:**

Let’s explore a scenario where there is a task involving the storage of a favourite number. For this purpose, we can start storing the variable `favoriteNumber` of type `uint`:

Uint256 favoriteNumber;

**Important:**  Every variable in Solidity comes with a default value. Uninitialized uint256 for example, defaults to `0` (zero) and an uninitialized boolean defaults to `false`.

**Conclusion:**

You’ve just filled in your first smart contract with variables and you explored the fundamental data types in Solidity.